

Plan de pruebas del proyecto integrado Aplicar Filtros

Los niveles de prueba que se van a aplicar son los siguientes:

- Pruebas unitarias. Estas pruebas tratan de verificar el comportamiento de clases de manera aislada, usando para ello el framework JUnit y, en caso de ser necesario simular el comportamiento de clases de las que se dependa, la librería Mockito.
- Pruebas de integración: Estas pruebas tratan de verificar la interacción entre diferentes clases del proyecto. Se llevarán a cabo usando el framework JUnit y la librería Mockito si es necesaria.
- Pruebas de interfaz de usuario (UI Tests): Estas pruebas se pueden ver cómo el nivel más avanzado de las pruebas de integración, cuando se prueban todos los componentes de la aplicación de manera conjunta. Se llevarán a cabo usando el framework Espresso.
- Pruebas de aceptación: Son las pruebas llevadas a cabo por los Product Owners durante el Sprint Review para comprobar si las historias de usuario elegidas para el Sprint se han realizado correctamente. Se ejecutarán de forma manual en el propio dispositivo del Product Owner u otro que se le proporcione. Las pruebas que se van a realizar deberán definirse en durante el Spring Initial Meeting.

A continuación, se muestra una especificación detallada de los casos de prueba a aplicar en cada nivel mencionado anteriormente para la historia de usuario *“AplicarFiltros”*.

PRUEBAS DE ACEPTACIÓN

Se identifican los siguientes escenarios:

PA00: Éxito.

1. El usuario selecciona la opción de aplicar filtro.
2. El sistema muestra la lista de gasolineras según los filtros seleccionados.
3. Se verifica que los filtros se han aplicado correctamente.

PA01: Error, no se aplican los filtros correctamente.

1. El usuario selecciona la opción de aplicar filtro.
2. El sistema muestra la lista de gasolineras errónea, no se aplican los filtros seleccionados.
3. Se verifica que las gasolineras no se muestran según los filtros seleccionados.

PA02: Error, se he intentado aplicar filtros de la misma categoría.

1. El usuario selecciona la opción de aplicar filtro, con la intención de escoger dos de la misma categoría.
2. Se muestra un mensaje de error al usuario que le informa de que, si escoge un filtro de la misma categoría, el otro quedará sin seleccionar.
3. El usuario acepta o rechaza los cambios.
4. Se verifica que el filtro que se ha elegido se aplica correctamente.

Los casos anteriormente definidos se encuentran resumidos en la Tabla 1.

Identificador	Entrada	Valor esperado
PA00	Un filtro	Lista filtrada
PA01	Un filtro	Lista mal filtrada
PA02	Dos filtros de la misma categoría (consecutivamente)	Lista filtrada

Tabla 1. Casos de prueba de aceptación.

PRUEBAS UNITARIAS

Los casos de prueba para el método ordena() del conjunto de clases de filtros. Ya que se ha aplicado una jerarquía de clases para construir los filtros, hay múltiples implementaciones de este método, pero se espera la misma salida general en todos. Además hay que probar que los filtros que son incompatibles no se pueden combinar, aunque para esto no se prueba ningún método en concreto, sino las interacciones entre ellos.

Identificador	Entrada	Valor esperado
UTD.1a	Lista original	Lista de gasolineras sin descuentos

Tabla 2. Casos de prueba válidos para el método ordena() de la clase SinDescuentoFiltro.

Identificador	Entrada	Valor esperado
UTD.1b	Lista original	Lista mal filtrada

Tabla 3. Casos de prueba no válidos para el método ordena () de la clase SinDescuentoFiltro.

Identificador	Entrada	Valor esperado
UTD.2a	Lista original	Lista de gasolineras con descuento

Tabla 4. Casos de prueba válidos para el método ordena() de la clase ConDescuentoFiltro.

Identificador	Entrada	Valor esperado
UTD.2b	Lista original	Lista mal filtrada

Tabla 5. Casos de prueba no válidos para el método ordena() de la clase ConDescuentoFiltro.

Identificador	Entrada	Valor esperado
UTD.3a	Lista original	Lista de gasolineras ordenadas de menor a mayor por precio de Diesel

Tabla 6. Casos de prueba válidos para el método ordena() de la clase DieselFiltro.

Identificador	Entrada	Valor esperado
UTD.3b	Lista original	Lista mal filtrada (desordenada)

Tabla 7. Casos de prueba no válidos para el método ordena() de la clase DieselFiltro.

Identificador	Entrada	Valor esperado
UTD.4a	Lista original	Lista de gasolineras con descuento ordenadas de menor a mayor por Gasolina95

Tabla 4. Casos de prueba válidos para el método ordena() de la clase Gasolina95Filtro.

Identificador	Entrada	Valor esperado
UTD.4b	Lista original	Lista mal filtrada

Tabla 5. Casos de prueba no válidos para el método ordena() de la clase Gasolina95Filtro.

Identificador	Entrada	Valor esperado
UTD.5a	Lista original → Filtro 95 → Filtro Gasoleo → Filtro Con Descuento	Lista de gasolineras con descuento ordenadas por gasóleo

Tabla 6. Caso de prueba para la compatibilidad de filtros. La flecha indica el orden en el que se aplican los filtros a la lista.

PRUEBAS DE INTEGRACIÓN

No hay para esta rama.

PRUEBAS DE INTERFAZ DE USUARIO

1. **ITIU00:** Una vez haya una serie de filtros aplicados, se comprobará que el nombre de los que están siendo mostrados en la interfaz como “seleccionados” coinciden con los filtros que se han aplicado a la lista.
2. **ITIU01:** Cuando se pulsa el botón de “Reset” de los filtros, se comprueba que la lista de filtros que se están aplicando a las gasolineras esta vacia

Daniel Sánchez Díez